UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/600,601	06/20/2003	Eric Anderson	100202201-1	5490

22879 7590 11/01/2007

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
SAEED, USMAAN	

ART UNIT	PAPER NUMBER
2166	

MAIL DATE	DELIVERY MODE
11/01/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/600,601
Filing Date: June 20, 2003
Appellant(s): ANDERSON, ERIC

Dan C. Hu
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 7/30/2007 appealing from the Office action mailed 4/27/2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after the office action of April 27, 2007 has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,381,619	Borowsky	4-2002

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-15, 17-22, 24-27, and 29-34 are rejected under 35 U.S.C. 102(b) as being anticipated by **Borowsky et al** (**Borowsky** hereinafter) (U.S. Patent 6,381,619).

With respect to claim 1, **Borowsky** teaches **a method for performing adaptive migration and execution, the method comprising:**

"obtaining a plan" as the migration plan generator develops a plan that leads to the lowest contention for the system (**Borowsky** Col 2, Lines 35-36). The reference teaches that the plan is being developed for migration. **"generated by a planner**

executable in a computer" as the present invention provides a computer data storage system with a migration plan generator which includes a "Simple" migration planner which provides for making terminal moves until no further terminal moves are possible based on random, preset, or functional ordering (**Borowsky** Col 2, Lines 15-20).

"adapting the plan to satisfy migration constraints" as the migration plan generator 100, the initial configuration data 110, the goal configuration data 112, and the set of constraints 114 are provided to a migration planner 116. The migration planner 116 either fails to develop a migration plan and exits to "fail" block 118 or produces a viable migration plan 120 (**Borowsky** Col 5, Lines 6-11). The plan being produced by the migration plan generator is only developed/adopted when the set of constraints provided to the planner are satisfied.

"executing at least one move of a data chunk in the plan" as the data stores are moved, or migrated, among the storage devices under the direction of a control 28 (**Borowsky** Col 3, Lines 49-51).

"feeding back information relating to the to the planner; and modifying the plan by the planner in response to the information" as (**Borowsky** Figure 5).

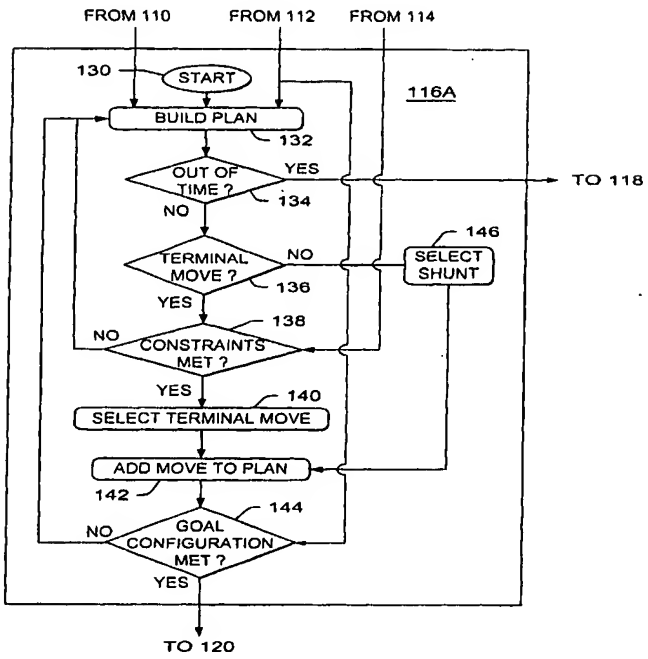

FIG. 5

Borowsky teaches once a terminal move is selected, it is added to the plan in an "add move to plan" block 142. The configuration after the added move is compared with the goal configuration data 112 in a "goal configuration met?" decision block 144. If the goal configuration is not met, the program returns to the "build plan" block 132, and if it is, it exits as the migration plan 120 (**Borowsky** Col 5, Lines 30-36). These lines and figure 5 teaches a feedback mechanism which send a feedback that goal configurations are not being met and to that to select another move. When a different/another move is being added to the plan, the plan is being modified.

With respect to claim 2, **Borowsky** further teaches **"the method of claim 1, wherein the steps in the method are repeated until no moves are pending"** as the present invention provides a computer data storage system with a migration plan generator which includes a "Simple" migration planner which provides for making terminal moves until no further terminal moves are possible based on random, preset, or functional ordering (**Borowsky** Col 2, Lines 15-20).

With respect to claim 3, **Borowsky** further teaches **"the method of claim 2, further comprising: waiting for all in-progress executions of moves to complete after no moves are pending"** as after a program has processed the possible moves, the then current configuration is compared with the goal configuration in the "goal configuration met?" decision block 172. If the goal configuration has not been met, the program proceeds back to the "select plan" block 152 and, if it has it, exits as the migration plan 120 (**Borowsky** Col 5, Lines 59-64). The reference teaches that the program processes/executes all the possible moves and it has to wait for all the executions in order to check if the configurations have been met.

With respect to claim 4, **Borowsky** further teaches **"the method of claim 1, further comprising: waiting for a move to complete if the adaptation of the plan indicates no moves meet the migration constraints"** as establishing that no other moves are possible after the store E move (**Borowsky** Col 6, Lines 47-48). The set of constraints 114 prevents certain moves (**Borowsky** Col 4, Line 54). This means that

after the completion of the store E move, no other moves are possible and constraints play a part in preventing certain moves.

With respect to claim 5, **Borowsky** further teaches **"the method of claim 1, further comprising:**

"estimating load value information" as the load placed by the move on the system should be minimized (in terms of data stores moved, time taken (parallel or sequential), bandwidth used, or similar metric) (**Borowsky** Col 6, Lines 24-27). **"using the load value information"** as the load placed by the move on the system should be minimized (in terms of data stores moved, time taken (parallel or sequential), bandwidth used, or similar metric) (**Borowsky** Col 6, Lines 24-27). **"assist in modifying the plan"** as an alternative migration plan would have been to move the store B from device 1 to the device 3 and move the store D from the device 2 to the device 1 (**Borowsky** Col 6, Lines 58-61). Figure 6 also have all the other different/modified/new plans that can be selected for use of migration. It is using the load value to find a modified/different/new plan since the constraints to select a plan contain a capacity constraint which examiner interprets as a load constraint.

With respect to claim 6, **Borowsky** further teaches **the method of claim 1, wherein adapting the plan comprises:**

selecting at least one step from the following

"pruning at least one move that violates a migration constraint" as the set of constraints 114 contains the capacities of the data storage devices, the capacities of the data stores, the bandwidth, movement rate, and other limitations on the moves (**Borowsky** Col 4, Lines 51-53). Therefore these set of constraints are being used in the pruning of the moves.

"selecting a largest set of moves that do not violates a migration constraint; and skipping a move that violates a migration constraint" as if a terminal move is possible, the program proceeds to the "constraints met?" decision block 138. The "constraints met?" decision block 138 receives the set of constraints 114 to determine whether or not the constraints have been met. If they have not been met, the program returns to the "build plan" block 132. If the constraints have been met, the program proceeds to the "select terminal move" block 140 (**Borowsky** Col 5, Lines 22-29). If the moves do not violate the migration constraints they are added to the plan and if they violate the constraints they are not added to that plan.

With respect to claim 7, **Borowsky** further teaches **"the method of claim 1, further comprising: treating a data chunk as existing in an old location and new location while a move is in progress"** as the initial configuration system 22, the initial configuration has device 1 with store A and store B assigned to it, device 2 with store C and store D assigned to it, and device 3 with store E assigned to it.
In the goal configuration system 26, the goal configuration has device 1 with store A, store D, and store E assigned to it, device 2 with store C and store B assigned to it, and

device 3 with no data block assigned to it (**Borowsky** Col 3, Lines 40-48). Therefore when a move is in progress it treats the initial and goal configuration, both having the data chunk E since they use memory in old and new locations.

With respect to claim 8, **Borowsky** further teaches **"the method of claim 1, further comprising: pruning moves that violate an access rule when a move is in progress, wherein the pruned moves are not selected for inclusion in the plan"** as there are different blocks of data in the storage system. The access patterns to these blocks of data changes over time. Further, devices may fail or be added or subtracted. Thus, the ultimate goal is a data storage system which is not only able to automatically configure itself, but to reconfigure itself `on-the-fly`; i.e. move stored data around based on changing access patterns (**Borowsky** Col 1 Lines 65-67, Col 2 Lines 1-4). The reference is pruning the moves that are violating the access rules since they are changed and then the reference is reconfiguring these moves based on the changed access patterns/rules. The pruned moved are not included unless they are changed and reconfigured.

With respect to claim 9, **Borowsky** further teaches **"the method of claim 7, wherein the step of treating the data chunk comprises: considering the data chunk as decreasing a per-node free space information at both the old location and the new location when a move is in progress"** as the present invention further provides a computer data storage system with a migration plan generator which

includes a "Greedy" migration planner which uses a "contention" metric. The "contention" of a data storage device is defined as the total size of the data stores that need to move onto such data storage device, divided by the amount of free space on such data storage device. The contention of the entire system is the sum of the contention over all the data storage devices. The migration plan generator develops a plan that leads to the lowest contention for the system (**Borowsky** Col 2, Lines 27-36). The free space is decreasing since plan generator s trying for develop a plan, which leads to lowest contention in order to use the least space required.

With respect to claim 10, **Borowsky** further teaches **a method for performing adaptive migration and execution, the method comprising:**

"obtaining a plan" as the migration plan generator develops a plan that leads to the lowest contention for the system (**Borowsky** Col 2, Lines 35-36). The reference teaches that the plan is being developed for migration. **"created by a planner executable in a computer"** as the present invention provides a computer data storage system with a migration plan generator which includes a "Simple" migration planner which provides for making terminal moves until no further terminal moves are possible based on random, preset, or functional ordering (**Borowsky** Col 2, Lines 15-20).

"determining all valid moves in the plan" as if there is a data storage device that has terminal moves going into it but none going out, then all these terminal moves will be valid, since the data storage device can clearly accommodate all the data stores

in the goal configuration (**Borowsky** Col 8, Lines 17-21). The reference is determining the valid moves that the data storage device can handle.

"executing a valid move" as the data stores are moved, or migrated, among the storage devices under the direction of a control 28 (**Borowsky** Col 3, Lines 49-51).

"if at least one additional move is required" as if there is a data storage device that has terminal moves going into it but none going out; then all these terminal moves will be valid, since the data storage device can clearly accommodate all the data stores in the goal configuration (**Borowsky** Col 8, Lines 17-21). The reference is determining the valid moves that the data storage device can handle and it would have additional moves after starting the first move.

"feeding back information relating to the to the planner; and modifying the plan by the planner in response to the information" as (**Borowsky** Figure 5).

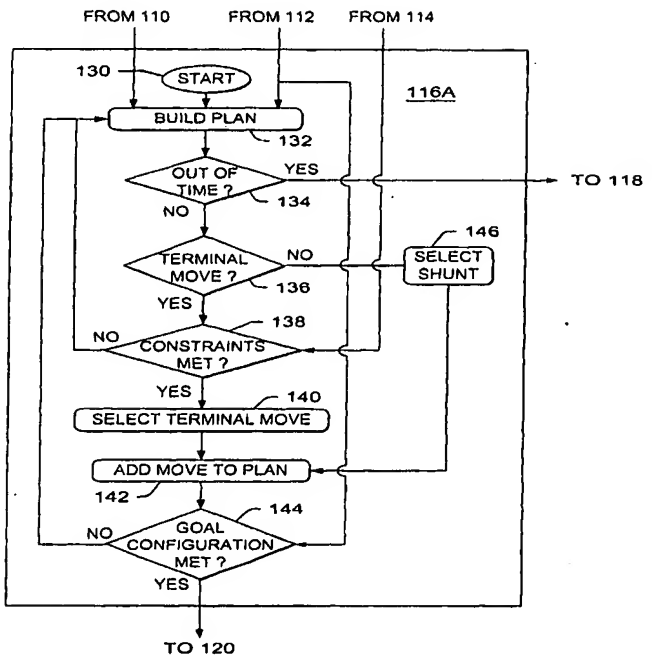

FIG. 5

Borowsky teaches once a terminal move is selected, it is added to the plan in an "add move to plan" block 142. The configuration after the added move is compared with the goal configuration data 112 in a "goal configuration met?" decision block 144. If the goal configuration is not met, the program returns to the "build plan" block 132, and if it is, it exits as the migration plan 120 (**Borowsky** Col 5, Lines 30-36). These lines and figure 5 teaches a feedback mechanism which send a feedback that goal configurations are not being met and to that to select another move. When a different/another move is being added to the plan, the plan is being modified.

With respect to claim 11, **Borowsky** further teaches **"the method of claim 10, further comprising: determining if an executor is available, wherein executing the valid more is performed by the available executor"** as after a program has processed the possible moves, the then current configuration is compared with the goal configuration in the "goal configuration met?" decision block 172. If the goal configuration has not been met, the program proceeds back to the "select plan" block 152 and, if it has it, exits as the migration plan 120 (**Borowsky** Col 5, Lines 59-64). The reference teaches that the program processes/executes all the possible moves and the executor has to be available to execute the moves.

Claim 12 is same as claim 2 and is rejected for the same reasons as applied hereinabove.

Claim 13 is same as claim 3 and is rejected for the same reasons as applied hereinabove.

With respect to claim 14, **Borowsky** further teaches **an article of manufacture, comprising: a machine-readable medium having stored thereon instructions to:**

"obtain a plan" as the migration plan generator develops a plan that leads to the lowest contention for the system (**Borowsky** Col 2, Lines 35-36). The reference teaches that the plan is being developed for migration.

"adapt the plan to satisfy migration constraints" as the migration plan generator 100, the initial configuration data 110, the goal configuration data 112, and the set of constraints 114 are provided to a migration planner 116. The migration planner 116 either fails to develop a migration plan and exits to "fail" block 118 or produces a viable migration plan 120 (**Borowsky** Col 5, Lines 6-11). The plan being produced by the migration plan generator is only developed/adopted when the set of constraints provided to the planner are satisfied.

"execute at least one move of a data chunk in the plan and executing another move" as the data stores are moved, or migrated, among the storage devices under the direction of a control 28 (**Borowsky** Col 4, Lines 49-51).

"modifying the plan based on feedback configuration information regarding in-progress execution of the at least one move" as (**Borowsky** Figure 5).

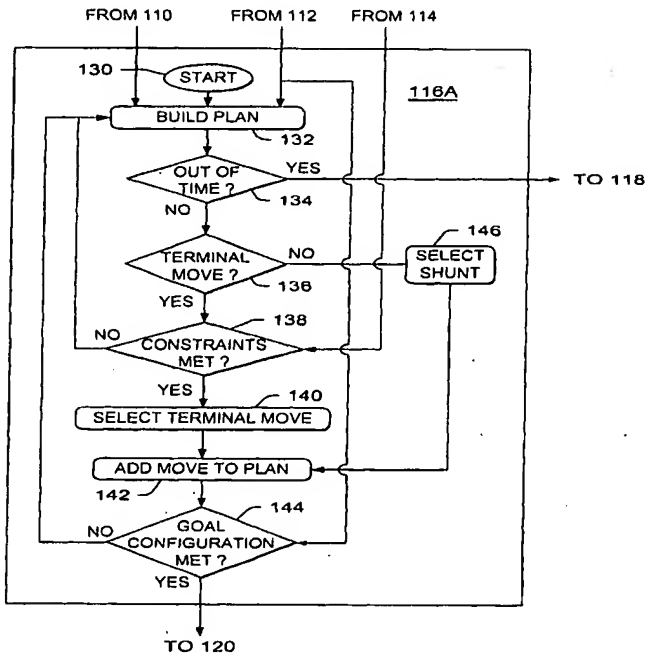

FIG. 5

Borowsky teaches once a terminal move is selected, it is added to the plan in an "add move to plan" block 142. The configuration after the added move is compared with the goal configuration data 112 in a "goal configuration met?" decision block 144. If the goal configuration is not met, the program returns to the "build plan" block 132, and if it is, it exits as the migration plan 120 (**Borowsky** Col 5, Lines 30-36). These lines and figure 5 teaches a feedback mechanism which send a feedback that goal configurations are not being met and to that to select another move. When a different/another move is being added to the plan, the plan is being modified.

With respect to claim 15, **Borowsky** further teaches **an apparatus for adaptive migration, the apparatus comprising:**

"a planner configured to generate a migration plan based upon configuration information" as the present invention provides a computer data storage system with a migration plan generator which includes a "Simple" migration planner which provides for making terminal moves until no further terminal moves are possible based on random, preset, or functional ordering (**Borowsky** Col 2, Lines 15-20). Fig 1 provides an overview of the invention and its migration from an initial configuration to a goal configuration (**Borowsky** Col 2, Lines 65-67). There is migration planner, which provides migration plan for making the moves. These moves are based on configuration information since initial configuration is being changed to final configuration.

"an adapter configured to receive the plan from the planner, to receive migration constraints information, target configuration information and current configuration information, and to transmit configuration information to the planner" as in the migration planner 116B of FIG. 6, the program begins at "start" block 150 and moves to select a plan in "select plan" block 152. The "select plan" block receives the initial configuration data 110, the goal configuration data 112, and the set of constraints 114 (**Borowsky** Col 8, Lines 36-40). The planner is receiving all the information about the initial/current configuration, goal/target configurations and the set of constraints.

"at least one executor configured to execute a move in the plan" as the plan might have to be executed in one hour. With a very slow data storage device, like a tape drive, and a large data store that needs to be moved, it might not be feasible to move the data store onto and off the tape drive because that may take more time than the hour set for completing the plan (**Borowsky** Col 5 Lines 1-5). Therefore the reference includes an executor, which executes a plan for moving data. **"wherein the configuration information relates to execution of the moves"** as after a program has processed the possible moves, the then current configuration is compared with the goal configuration in the "goal configuration met?" decision block 172. If the goal configuration has not been met, the program proceeds back to the "select plan" block 152 and, if it has it, exits as the migration plan 120 (**Borowsky** Col 5, Lines 59-64 and Figures 2 and 5).

Claim 16 (Cancelled).

Claim 17, 18, and 19 are essentially the same as claim 5 except they set forth the claimed invention as an apparatus and are rejected for the same reasons as applied hereinabove.

With respect to claim 20, **Borowsky** further teaches **"the apparatus of claim 15, wherein the adapter iteratively obtains plans from the planner until no moves are pending"** as after a program has processed the possible moves, the then current

configuration is compared with the goal configuration in the "goal configuration met?" decision block 172. If the goal configuration has not been met, the program proceeds back to the "select plan" block 152 and, if it has it, exits as the migration plan 120 (**Borowsky** Col 5, Lines 59-64). The reference keeps on selecting a different plan until the configuration/(moves) have been met.

The present invention further provides a computer data storage system with a migration plan generator which includes a "Meta" migration planner which shifts from one planner to another based on the planner's performance (**Borowsky** Col 2, Lines 55-68).

Claim 21 is essentially the same as claim 3 except it sets forth the claimed invention as an apparatus and is rejected for the same reasons as applied hereinabove.

Claim 22 is essentially the same as claim 4 except it sets forth the claimed invention as an apparatus and is rejected for the same reasons as applied hereinabove.

Claim 23 (Cancelled).

Claim 24 is essentially the same as claim 6 except it sets forth the claimed invention as an apparatus and is rejected for the same reasons as applied hereinabove.

Claim 25 is essentially the same as claim 7 except it sets forth the claimed invention as an apparatus and is rejected for the same reasons as applied hereinabove.

Claim 26 is essentially the same as claim 8 except it sets forth the claimed invention as an apparatus and is rejected for the same reasons as applied hereinabove.

Claim 27 is essentially the same as claim 9 except it sets forth the claimed invention as an apparatus and is rejected for the same reasons as applied hereinabove.

Claim 28 (Cancelled).

With respect to claim 29, **Borowsky** teaches **the method of claim 1, further comprising:**

"executing at least a second move of a data chunk" as after a program has processed the possible moves, the then current configuration is compared with the goal configuration in the "goal configuration met?" decision block 172. If the goal configuration has not been met, the program proceeds back to the "select plan" block 152 and, if it has it, exits as the migration plan 120 (**Borowsky** Col 5, Lines 59-64). The present invention further provides a computer data storage system with a migration plan generator which includes a "Meta" migration planner which shifts from one planner to another based on the planner's performance (**Borowsky** Col 2, Lines 55-68).

"feeding back information relating to the execution of the at least second move to the planner" as (**Borowsky** Figure 5).

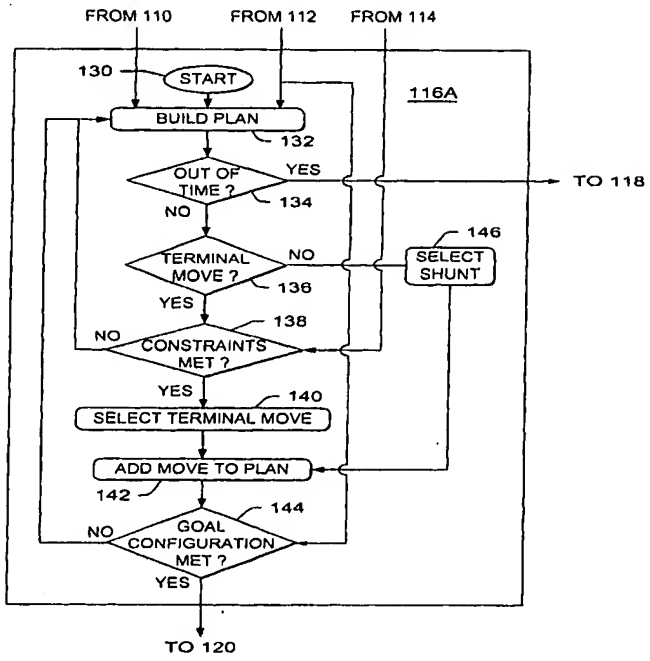

FIG. 5

Borowsky teaches once a terminal move is selected, it is added to the plan in an "add move to plan" block 142. The configuration after the added move is compared with the goal configuration data 112 in a "goal configuration met?" decision block 144. If the goal configuration is not met, the program returns to the "build plan" block 132, and if it is, it exits as the migration plan 120 (**Borowsky** Col 5, Lines 30-36). These lines and figure 5 teaches a feedback mechanism which send a feedback that goal configurations are not being met and to that to select another move. When a different/another move is being added to the plan, the plan is being modified.

The present invention further provides a computer data storage system with a migration plan generator which includes a "Meta" migration planner which shifts from one planner to another based on the planner's performance (**Borowsky** Col 2, Lines 55-68).

"further modifying the plan by the planner in response to the information relating to the execution of the at least second move" as (**Borowsky** Figure 5).

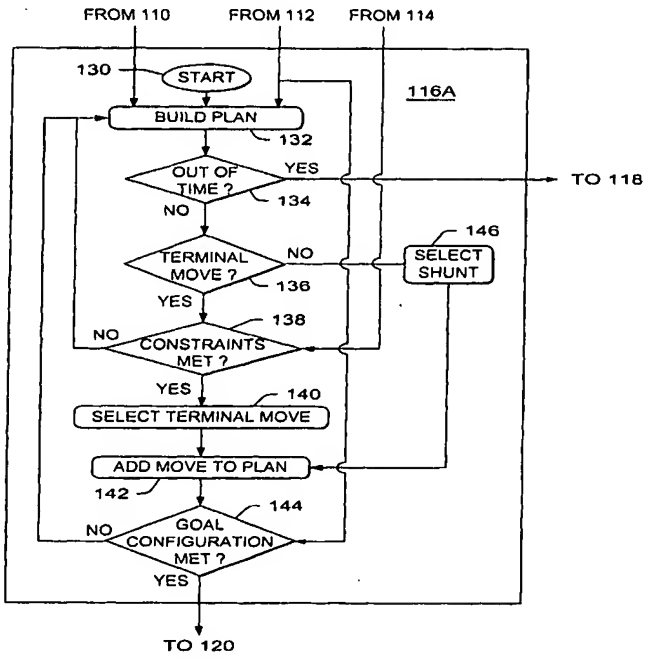

FIG. 5

Borowsky teaches once a terminal move is selected, it is added to the plan in an "add move to plan" block 142. The configuration after the added move is compared with the goal configuration data 112 in a "goal configuration met?" decision block 144. If

the goal configuration is not met, the program returns to the "build plan" block 132, and if it is, it exits as the migration plan 120 (**Borowsky** Col 5, Lines 30-36). These lines and figure 5 teaches a feedback mechanism which send a feedback that goal configurations are not being met and to that to select another move. When a different/another move is being added to the plan, the plan is being modified.

The present invention further provides a computer data storage system with a migration plan generator which includes a "Meta" migration planner which shifts from one planner to another based on the planner's performance (**Borowsky** Col 2, Lines 55-68).

With respect to claim 30, **Borowsky** teaches **the method of claim 1, wherein execution of the at least one move is performed by an executor, the method further comprising:**

"waiting for the executor to complete the at least one move"

"determining whether another move is to be executed" as the data stores are moved, or migrated, among the storage devices under the direction of a control 28 (**Borowsky** Col 3, Lines 49-51). After a program has processed the possible moves, the then current configuration is compared with the goal configuration in the "goal configuration met?" decision block 172. If the goal configuration has not been met, the program proceeds back to the "select plan" block 152 and, if it has it, exits as the migration plan 120 (**Borowsky** Col 5, Lines 59-64). The present invention further provides a computer data storage system with a migration plan generator which

includes a "Meta" migration planner which shifts from one planner to another based on the planner's performance (**Borowsky** Col 2, Lines 55-68). These lines teach that all the possible/determined move are being performed.

"wherein modifing the plan is performed in response to determining that the another move is to be executed" as (**Borowsky** Figure 5).

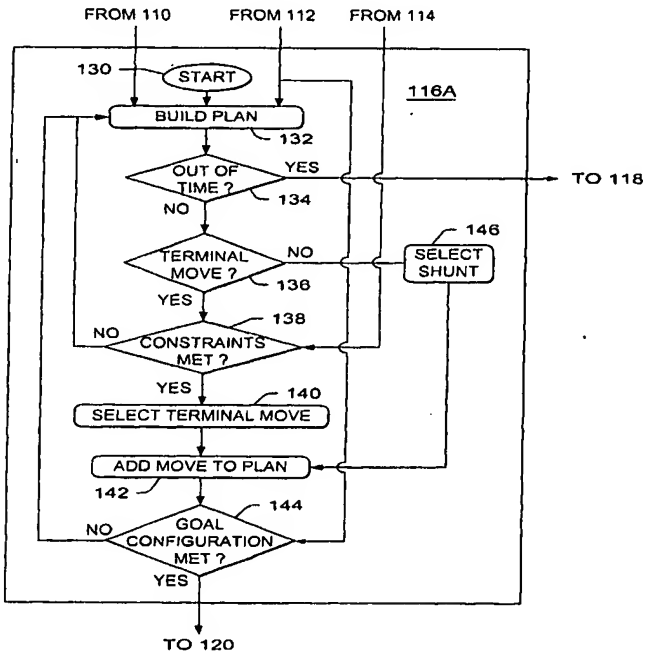

FIG. 5

Borowsky teaches once a terminal move is selected, it is added to the plan in an "add move to plan" block 142. The configuration after the added move is compared with the goal configuration data 112 in a "goal configuration met?" decision block 144. If the goal configuration is not met, the program returns to the "build plan" block 132, and

if it is, it exits as the migration plan 120 (**Borowsky** Col 5, Lines 30-36). These lines and figure 5 teaches a feedback mechanism which send a feedback that goal configurations are not being met and to that to select another move. When a different/another move is being added to the plan, the plan is being modified.

Claim 32 is essentially the same as claim 30 except it sets forth the claimed invention as an article of manufacture and is rejected for the same reasons as applied hereinabove.

With respect to claim 31, **Borowsky** teaches **the method of claim 1, further comprising:**

"tracking the information relating to the execution of the at least one move by an adapter that also adapts the plan to satisfy migration constraints" as the data stores are moved, or migrated, among the storage devices under the direction of a control 28 (**Borowsky** Col 3, Lines 49-51). After a program has processed the possible moves, the then current configuration is compared with the goal configuration in the "goal configuration met?" decision block 172. If the goal configuration has not been met, the program proceeds back to the "select plan" block 152 and, if it has it, exits as the migration plan 120 (**Borowsky** Col 5, Lines 59-64). The present invention further provides a computer data storage system with a migration plan generator which includes a "Meta" migration planner which shifts from one planner to another based on the planner's performance (**Borowsky** Col 2, Lines 55-68).

The migration planner 116B of FIG. 6, the program begins at "start" block 150 and moves to select a plan in "select plan" block 152. The "select plan" block receives the initial configuration data 110, the goal configuration data 112, and the set of constraints 114 (**Borowsky** Col 8, Lines 36-40).

"wherein feeding back the information is performed by the adapter to the planner" as (**Borowsky** Figure 5).

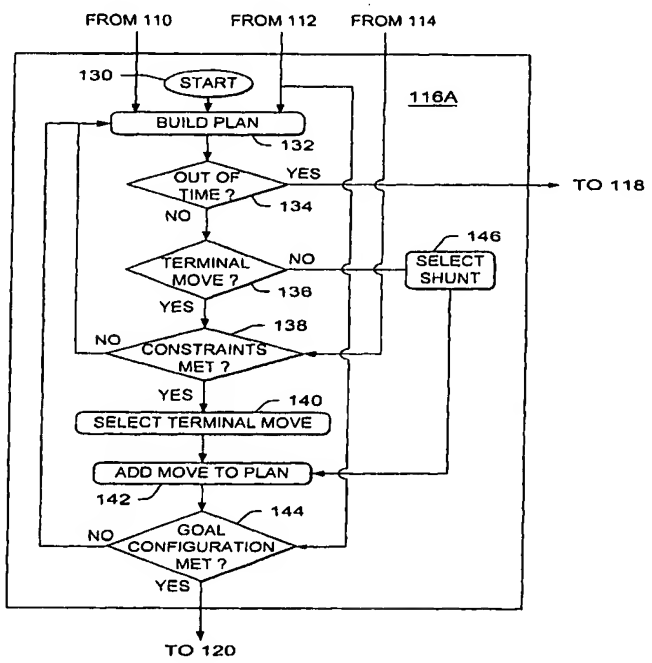

FIG. 5

Borowsky teaches once a terminal move is selected, it is added to the plan in an "add move to plan" block 142. The configuration after the added move is compared with the goal configuration data 112 in a "goal configuration met?" decision block 144. If

the goal configuration is not met, the program returns to the "build plan" block 132, and if it is, it exits as the migration plan 120 (**Borowsky** Col 5, Lines 30-36). These lines and figure 5 teaches a feedback mechanism which send a feedback that goal configurations are not being met and to that to select another move. When a different/another mover is being added to the plan, the plan is being modified.

With respect to claim 33, **Borowsky** teaches **the article of claim 14, wherein the machine-readable medium further contains instructions to:**

"estimate load information associated with the plan" as the load placed by the move on the system should be minimized (in terms of data stores moved, time taken (parallel or sequential), bandwidth used, or similar metric) (**Borowsky** Col 6, Lines 24-27).

"wherein modifying the plan is further based on the estimated load information" as the load placed by the move on the system should be minimized (in terms of data stores moved, time taken (parallel or sequential), bandwidth used, or similar metric) (**Borowsky** Col 6, Lines 24-27). An alternative migration plan would have been to move the store B from device 1 to the device 3 and move the store D from the device 2 to the device 1 (**Borowsky** Col 6, Lines 58-61). Figure 6 also have all the other different/modified/new plans that can be selected for use of migration. It is using the load value to find a modified/different/new plan since the constraints to select a plan contain a capacity constraint which examiner interprets as a load constraint.

Claim 34 is essentially the same as claims 31 and 20 and is rejected for the same reason as applied hereinabove.

(10) Response to Argument

A. § 102(b) rejection of claims 1-15, 17-22, 24-27, and 29-34 over Borowsky.

Appellant argues that **Borowsky** does not teach or suggest **"feeding back information relating to the execution of the at least one move to the planner; and modifying the plan by the planner in response to the information"** as required by independent claims 1 and 10, **"modifying the plan based on feedback configuration information regarding in-progress execution of the at least one move"** as required by independent claim 14 and **"a planner that is able to generate a configuration plan based upon configuration information that related to execution of a move"** as required by independent claim 15.

In response to the preceding arguments examiner respectfully submits that **Borowsky** teaches **"feeding back information relating to the execution of the at least one move to the planner; and modifying the plan by the planner in response to the information"** as (**Borowsky** Figure 5).

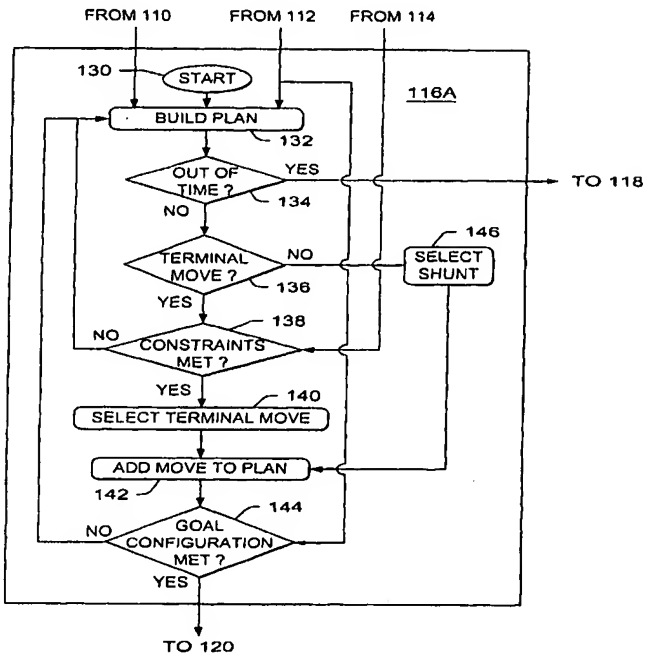

FIG. 5

Borowsky teaches once a terminal move is selected, it is added to the plan in an "add move to plan" block 142. The configuration after the added move is compared with the goal configuration data 112 in a "goal configuration met?" decision block 144. If the goal configuration is not met, the program returns to the "build plan" block 132, and if it is, it exits as the migration plan 120 (**Borowsky** Col 5, Lines 30-36).

Further, **Borowsky** teaches the data stores are moved, or migrated, among the storage devices under the direction of a control 28 such as a computer, processor, controller, etc (**Borowsky** Col 3, Lines 49-51). The set of constraints 114 contains the

capacities of the data storage devices, the capacities of the data stores, the bandwidth, movement rate, and other limitations on the moves (**Borowsky** Col 4, Lines 50-53).

These lines and figure 5 teaches a feedback mechanism which send a feedback that goal configurations are not being met and to select another move. When a different/another move is being added to the plan, the plan is being modified. This feedback information is relating to the execution of a move because when control 28 such as a computer/processor is making a move/executing, the constraints 138 (the capacities of the data storage devices; the capacities of the data stores, the bandwidth, movement rate, and other limitations on the moves) are being checked which all related to the execution of the moves.

Therefore, Borowsky is providing the feedback information related to the execution of at least one move.

Further, **Borowsky** teaches, **"modifying the plan based on feedback configuration information regarding in-progress execution of the at least one move"** as (**Borowsky** Figure 5).

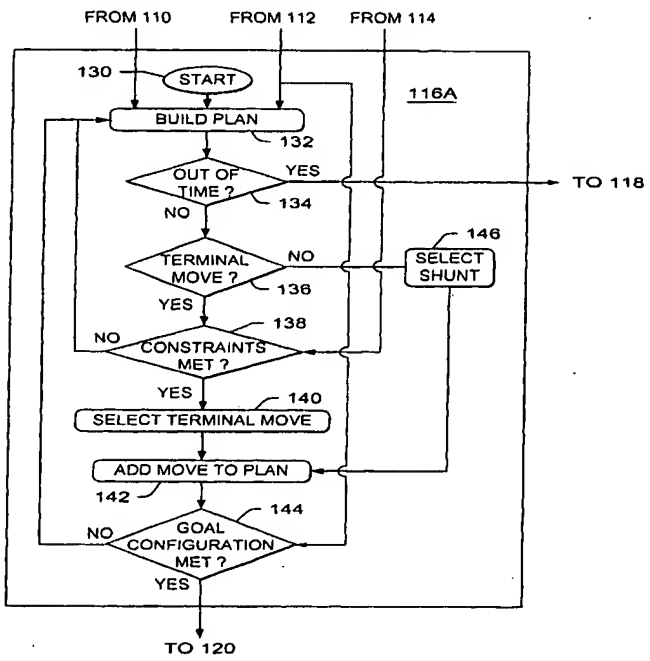

FIG. 5

Borowsky teaches once a terminal move is selected, it is added to the plan in an "add move to plan" block 142. The configuration after the added move is compared with the goal configuration data 112 in a "goal configuration met?" decision block 144. If the goal configuration is not met, the program returns to the "build plan" block 132, and if it is, it exits as the migration plan 120 (**Borowsky** Col 5, Lines 30-36).

Further, **Borowsky** teaches the data stores are moved, or migrated, among the storage devices under the direction of a control 28 such as a computer, processor, controller, etc (**Borowsky** Col 3, Lines 49-51). The set of constraints 114 contains the

capacities of the data storage devices, the capacities of the data stores, the bandwidth, movement rate, and other limitations on the moves (**Borowsky** Col 4, Lines 50-53).

These lines and figure 5 teaches a feedback mechanism which send a feedback that goal configurations are not being met and to select another move. When a different/another move is being added to the plan, the plan is being modified. This feedback information is relating to the in-progress execution of a move because when control 28 such as a computer/processor is making a move/executing, the constraints 138 (the capacities of the data storage devices, the capacities of the data stores, the bandwidth, movement rate, and other limitations on the moves) are being checked which all related to the execution of the moves.

Therefore, Borowsky is providing the feedback information related to the in-progress execution of at least one move.

Further, **Borowsky** teaches, **"a planner that is able to generate a configuration plan based upon configuration information that related to execution of a move"** as (**Borowsky** Figure 5).

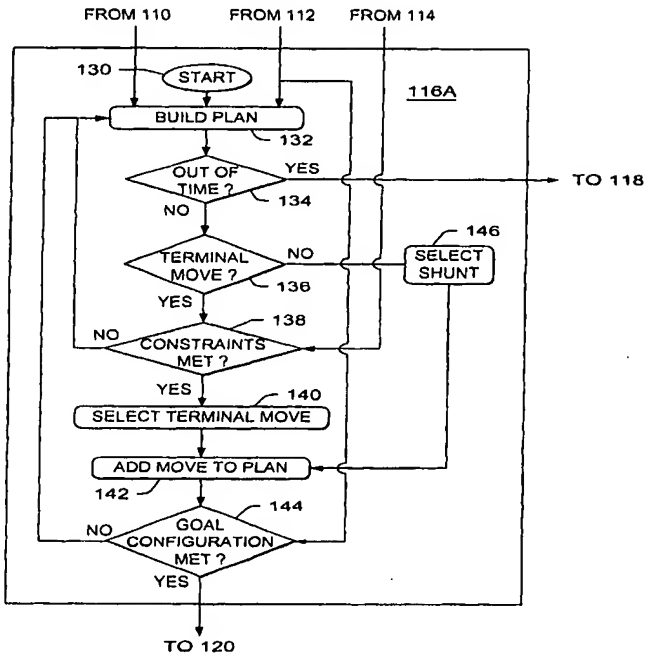

FIG. 5

Borowsky teaches once a terminal move is selected, it is added to the plan in an "add move to plan" block 142. The configuration after the added move is compared with the goal configuration data 112 in a "goal configuration met?" decision block 144. If the goal configuration is not met, the program returns to the "build plan" block 132, and if it is, it exits as the migration plan 120 (**Borowsky** Col 5, Lines 30-36).

Further, **Borowsky** teaches the data stores are moved, or migrated, among the storage devices under the direction of a control 28 such as a computer, processor, controller, etc (**Borowsky** Col 3, Lines 49-51). The set of constraints 114 contains the

capacities of the data storage devices, the capacities of the data stores, the bandwidth, movement rate, and other limitations on the moves (**Borowsky** Col 4, Lines 50-53).

These lines and figure 5 teaches a feedback mechanism which send a feedback that goal configurations are not being met and to select another move. When a different/another move is being added to the plan, the plan is being modified. This feedback information is relating to the execution of a move because when control 28 such as a computer/processor is making a move/executing, the constraints 138 (the capacities of the data storage devices, the capacities of the data stores, the bandwidth, movement rate, and other limitations on the moves) are being checked which all related to the execution of the moves.

Therefore, Borowsky is providing the feedback information related to the execution of at least one move and is building a configuration plan which is related to the execution of at least one move.

Appellant's arguments directed towards the rejections of dependent claim 2-9, 11-13, 17-22, 24-27, and 29-34 reiterate deficiencies Appellant made in the rejection of the independent claims 1, 10, 14 and 15 and do not address any new points. Therefore examiner submits that if the rejection of the independent claims is deemed proper, the rejection of claims 2-9, 11-13, 17-22, 24-27, and 29-34 should also be upheld.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Usmaan Saeed

Examiner

Conferees:

Hosain Alam

Supervisory Patent Examiner

HOSAIN ALAM
SUPERVISORY PATENT EXAMINER

Eddie Lee

TQAS/Appeals specialist

EDDIE C. LEE
SUPERVISORY PATENT EXAMINER